| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicOptionPaneUI.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicMenuUI.MouseInputHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.ButtonActionListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicOptionPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicOptionPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

## **javax.swing.plaf.basic**

Class BasicOptionPaneUI

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.plaf.ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)  
 [javax.swing.plaf.OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html)  
 **javax.swing.plaf.basic.BasicOptionPaneUI**

public class **BasicOptionPaneUI**extends [OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html)

Provides the basic look and feel for a JOptionPane. BasicMessagePaneUI provides a means to place an icon, message and buttons into a Container. Generally, the layout will look like:

------------------  
 | i | message |  
 | c | message |  
 | o | message |  
 | n | message |  
 ------------------  
 | buttons |  
 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

icon is an instance of Icon that is wrapped inside a JLabel. The message is an opaque object and is tested for the following: if the message is a Component it is added to the Container, if it is an Icon it is wrapped inside a JLabel and added to the Container otherwise it is wrapped inside a JLabel.

The above layout is used when the option pane's ComponentOrientation property is horizontal, left-to-right. The layout will be adjusted appropriately for other orientations.

The Container, message, icon, and buttons are all determined from abstract methods.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicOptionPaneUI.ButtonActionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.ButtonActionListener.html)            This inner class is marked "public" due to a compiler bug. |
| static class | [**BasicOptionPaneUI.ButtonAreaLayout**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.ButtonAreaLayout.html)            ButtonAreaLayout behaves in a similar manner to FlowLayout. |
| class | [**BasicOptionPaneUI.PropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.PropertyChangeHandler.html)            This inner class is marked "public" due to a compiler bug. |

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**hasCustomComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#hasCustomComponents)            This is set to true in validateComponent if a Component is contained in either the message or the buttons. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**initialFocusComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#initialFocusComponent)            Component to receive focus when messaged with selectInitialValue. |
| protected  [JComponent](http://docs.google.com/javax/swing/JComponent.html) | [**inputComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#inputComponent)            JComponent provide for input if optionPane.getWantsInput() returns true. |
| static int | [**MinimumHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#MinimumHeight) |
| protected  [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#minimumSize) |
| static int | [**MinimumWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#MinimumWidth) |
| protected  [JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) | [**optionPane**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#optionPane)            JOptionPane that the receiver is providing the look and feel for. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**propertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#propertyChangeListener) |

| **Constructor Summary** | |
| --- | --- |
| [**BasicOptionPaneUI**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#BasicOptionPaneUI())() |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addButtonComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#addButtonComponents(java.awt.Container,%20java.lang.Object%5B%5D,%20int))([Container](http://docs.google.com/java/awt/Container.html) container, [Object](http://docs.google.com/java/lang/Object.html)[] buttons, int initialIndex)            Creates the appropriate object to represent each of the objects in buttons and adds it to container. |
| protected  void | [**addIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#addIcon(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) top)            Creates and adds a JLabel representing the icon returned from getIcon to top. |
| protected  void | [**addMessageComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#addMessageComponents(java.awt.Container,%20java.awt.GridBagConstraints,%20java.lang.Object,%20int,%20boolean))([Container](http://docs.google.com/java/awt/Container.html) container, [GridBagConstraints](http://docs.google.com/java/awt/GridBagConstraints.html) cons, [Object](http://docs.google.com/java/lang/Object.html) msg, int maxll, boolean internallyCreated)            Creates the appropriate object to represent msg and places it into container. |
| protected  void | [**burstStringInto**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#burstStringInto(java.awt.Container,%20java.lang.String,%20int))([Container](http://docs.google.com/java/awt/Container.html) c, [String](http://docs.google.com/java/lang/String.html) d, int maxll)            Recursively creates new JLabel instances to represent d. |
| boolean | [**containsCustomComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#containsCustomComponents(javax.swing.JOptionPane))([JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) op)            Returns true if in the last call to validateComponent the message or buttons contained a subclass of Component. |
| protected  [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) | [**createButtonActionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createButtonActionListener(int))(int buttonIndex) |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**createButtonArea**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createButtonArea())()            Creates and returns a Container containing the buttons. |
| protected  [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) | [**createLayoutManager**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createLayoutManager())() |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**createMessageArea**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createMessageArea())()            Messaged from installComponents to create a Container containing the body of the message. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createPropertyChangeListener())() |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**createSeparator**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createSeparator())() |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) x)            Creates a new BasicOptionPaneUI instance. |
| protected  [Object](http://docs.google.com/java/lang/Object.html)[] | [**getButtons**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getButtons())()            Returns the buttons to display from the JOptionPane the receiver is providing the look and feel for. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getIcon())()            Returns the icon from the JOptionPane the receiver is providing the look and feel for, or the default icon as returned from getDefaultIcon. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIconForType**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getIconForType(int))(int messageType)            Returns the icon to use for the passed in type. |
| protected  int | [**getInitialValueIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getInitialValueIndex())()            Returns the initial index into the buttons to select. |
| protected  int | [**getMaxCharactersPerLineCount**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getMaxCharactersPerLineCount())()            Returns the maximum number of characters to place on a line. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**getMessage**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getMessage())()            Returns the message to display from the JOptionPane the receiver is providing the look and feel for. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumOptionPaneSize**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getMinimumOptionPaneSize())()            Returns the minimum size the option pane should be. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            If c is the JOptionPane the receiver is contained in, the preferred size that is returned is the maximum of the preferred size of the LayoutManager for the JOptionPane, and getMinimumOptionPaneSize. |
| protected  boolean | [**getSizeButtonsToSameWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#getSizeButtonsToSameWidth())()            Returns true, basic L&F wants all the buttons to have the same width. |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#installComponents())() |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#installDefaults())() |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#installKeyboardActions())() |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#installListeners())() |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Installs the receiver as the L&F for the passed in JOptionPane. |
| protected  void | [**resetInputValue**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#resetInputValue())()            Sets the input value in the option pane the receiver is providing the look and feel for based on the value in the inputComponent. |
| void | [**selectInitialValue**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#selectInitialValue(javax.swing.JOptionPane))([JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) op)            If inputComponent is non-null, the focus is requested on that, otherwise request focus on the default value |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#uninstallComponents())() |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#uninstallDefaults())() |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#uninstallKeyboardActions())() |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Removes the receiver from the L&F controller of the passed in split pane. |

| **Methods inherited from class javax.swing.plaf.**[**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html) |
| --- |
| [contains](http://docs.google.com/javax/swing/plaf/ComponentUI.html#contains(javax.swing.JComponent,%20int,%20int)), [getAccessibleChild](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int)), [getAccessibleChildrenCount](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent)), [getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)), [getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)), [getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)), [paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)), [update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### MinimumWidth

public static final int **MinimumWidth**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicOptionPaneUI.MinimumWidth)

### MinimumHeight

public static final int **MinimumHeight**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicOptionPaneUI.MinimumHeight)

### optionPane

protected [JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) **optionPane**

JOptionPane that the receiver is providing the look and feel for.

### minimumSize

protected [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumSize**

### inputComponent

protected [JComponent](http://docs.google.com/javax/swing/JComponent.html) **inputComponent**

JComponent provide for input if optionPane.getWantsInput() returns true.

### initialFocusComponent

protected [Component](http://docs.google.com/java/awt/Component.html) **initialFocusComponent**

Component to receive focus when messaged with selectInitialValue.

### hasCustomComponents

protected boolean **hasCustomComponents**

This is set to true in validateComponent if a Component is contained in either the message or the buttons.

### propertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **propertyChangeListener**

| **Constructor Detail** |
| --- |

### BasicOptionPaneUI

public **BasicOptionPaneUI**()

| **Method Detail** |
| --- |

### createUI

public static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **createUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) x)

Creates a new BasicOptionPaneUI instance.

### installUI

public void **installUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Installs the receiver as the L&F for the passed in JOptionPane.

**Overrides:**[installUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component where this UI delegate is being installed**See Also:**[ComponentUI.uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)), [JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### uninstallUI

public void **uninstallUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Removes the receiver from the L&F controller of the passed in split pane.

**Overrides:**[uninstallUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component from which this UI delegate is being removed; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### installDefaults

protected void **installDefaults**()

### uninstallDefaults

protected void **uninstallDefaults**()

### installComponents

protected void **installComponents**()

### uninstallComponents

protected void **uninstallComponents**()

### createLayoutManager

protected [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **createLayoutManager**()

### installListeners

protected void **installListeners**()

### uninstallListeners

protected void **uninstallListeners**()

### createPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createPropertyChangeListener**()

### installKeyboardActions

protected void **installKeyboardActions**()

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

### getMinimumOptionPaneSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumOptionPaneSize**()

Returns the minimum size the option pane should be. Primarily provided for subclassers wishing to offer a different minimum size.

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

If c is the JOptionPane the receiver is contained in, the preferred size that is returned is the maximum of the preferred size of the LayoutManager for the JOptionPane, and getMinimumOptionPaneSize.

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose preferred size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### createMessageArea

protected [Container](http://docs.google.com/java/awt/Container.html) **createMessageArea**()

Messaged from installComponents to create a Container containing the body of the message. The icon is the created by calling addIcon.

### addMessageComponents

protected void **addMessageComponents**([Container](http://docs.google.com/java/awt/Container.html) container,  
 [GridBagConstraints](http://docs.google.com/java/awt/GridBagConstraints.html) cons,  
 [Object](http://docs.google.com/java/lang/Object.html) msg,  
 int maxll,  
 boolean internallyCreated)

Creates the appropriate object to represent msg and places it into container. If msg is an instance of Component, it is added directly, if it is an Icon, a JLabel is created to represent it, otherwise a JLabel is created for the string, if d is an Object[], this method will be recursively invoked for the children. internallyCreated is true if Objc is an instance of Component and was created internally by this method (this is used to correctly set hasCustomComponents only if !internallyCreated).

### getMessage

protected [Object](http://docs.google.com/java/lang/Object.html) **getMessage**()

Returns the message to display from the JOptionPane the receiver is providing the look and feel for.

### addIcon

protected void **addIcon**([Container](http://docs.google.com/java/awt/Container.html) top)

Creates and adds a JLabel representing the icon returned from getIcon to top. This is messaged from createMessageArea

### getIcon

protected [Icon](http://docs.google.com/javax/swing/Icon.html) **getIcon**()

Returns the icon from the JOptionPane the receiver is providing the look and feel for, or the default icon as returned from getDefaultIcon.

### getIconForType

protected [Icon](http://docs.google.com/javax/swing/Icon.html) **getIconForType**(int messageType)

Returns the icon to use for the passed in type.

### getMaxCharactersPerLineCount

protected int **getMaxCharactersPerLineCount**()

Returns the maximum number of characters to place on a line.

### burstStringInto

protected void **burstStringInto**([Container](http://docs.google.com/java/awt/Container.html) c,  
 [String](http://docs.google.com/java/lang/String.html) d,  
 int maxll)

Recursively creates new JLabel instances to represent d. Each JLabel instance is added to c.

### createSeparator

protected [Container](http://docs.google.com/java/awt/Container.html) **createSeparator**()

### createButtonArea

protected [Container](http://docs.google.com/java/awt/Container.html) **createButtonArea**()

Creates and returns a Container containing the buttons. The buttons are created by calling getButtons.

### addButtonComponents

protected void **addButtonComponents**([Container](http://docs.google.com/java/awt/Container.html) container,  
 [Object](http://docs.google.com/java/lang/Object.html)[] buttons,  
 int initialIndex)

Creates the appropriate object to represent each of the objects in buttons and adds it to container. This differs from addMessageComponents in that it will recurse on buttons and that if button is not a Component it will create an instance of JButton.

### createButtonActionListener

protected [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) **createButtonActionListener**(int buttonIndex)

### getButtons

protected [Object](http://docs.google.com/java/lang/Object.html)[] **getButtons**()

Returns the buttons to display from the JOptionPane the receiver is providing the look and feel for. If the JOptionPane has options set, they will be provided, otherwise if the optionType is YES\_NO\_OPTION, yesNoOptions is returned, if the type is YES\_NO\_CANCEL\_OPTION yesNoCancelOptions is returned, otherwise defaultButtons are returned.

### getSizeButtonsToSameWidth

protected boolean **getSizeButtonsToSameWidth**()

Returns true, basic L&F wants all the buttons to have the same width.

### getInitialValueIndex

protected int **getInitialValueIndex**()

Returns the initial index into the buttons to select. The index is calculated from the initial value from the JOptionPane and options of the JOptionPane or 0.

### resetInputValue

protected void **resetInputValue**()

Sets the input value in the option pane the receiver is providing the look and feel for based on the value in the inputComponent.

### selectInitialValue

public void **selectInitialValue**([JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) op)

If inputComponent is non-null, the focus is requested on that, otherwise request focus on the default value

**Specified by:**[selectInitialValue](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html#selectInitialValue(javax.swing.JOptionPane)) in class [OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html)

### containsCustomComponents

public boolean **containsCustomComponents**([JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) op)

Returns true if in the last call to validateComponent the message or buttons contained a subclass of Component.

**Specified by:**[containsCustomComponents](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html#containsCustomComponents(javax.swing.JOptionPane)) in class [OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicOptionPaneUI.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicMenuUI.MouseInputHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicOptionPaneUI.ButtonActionListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicOptionPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicOptionPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).